Nonstationary EO/IR Clutter Suppression and Dim Object Tracking

Alexander G. Tartakovsky
Argo Science Corp. and University of Southern California
Los Angeles, CA

Andrew P. Brown
Toyon Research Corporation
Goleta, CA

James Brown
Air Force Research Laboratory
Hanscom AFB, MA

ABSTRACT

We develop and evaluate the performance of advanced algorithms which provide significantly improved capabilities for automated detection and tracking of ballistic and flying dim objects in the presence of highly structured intense clutter. Applications include ballistic missile early warning, midcourse tracking, trajectory prediction, and resident space object detection and tracking. The set of algorithms include, in particular, adaptive spatiotemporal clutter estimation-suppression and nonlinear filtering-based multiple-object track-before-detect. These algorithms are suitable for integration into geostationary, highly elliptical, or low earth orbit scanning or staring sensor suites, and are based on data-driven processing that adapts to real-world clutter backgrounds, including celestial, earth limb, or terrestrial clutter. In many scenarios of interest, e.g., for highly elliptic and, especially, low earth orbits, the resulting clutter is highly nonstationary, providing a significant challenge for clutter suppression to or below sensor noise levels, which is essential for dim object detection and tracking. We demonstrate the success of the developed algorithms using semi-synthetic and real data. In particular, our algorithms are shown to be capable of detecting and tracking point objects with signal-to-clutter levels down to 1/1000 and signal-to-noise levels down to 1/4.

1. INTRODUCTION

We are interested in the efficient tracking of multiple small objects using space-based EO/IR sensors. This is a challenging problem since objects are usually very dim with respect to the cluttered image background and often even with respect to sensor noise. For example, the signal-to-clutter ratio (SCR) may be as low as 10^{-3}. This problem has been previously discussed by Tartakovsky and Brown [7] for quasi-stationary conditions, e.g., for geostationary (GEO) platforms, and by Tartakovsky, Brown, and Brown [8] in the 2009 AMOS conference for both GEO and nonstationary platforms, such as low earth orbits (LEO) and highly elliptic orbits (HEO). As has been shown in these works, successful detection and tracking can usually be achieved only when applying sophisticated spatiotemporal image processing that allows for an accurate Clutter Estimation and Rejection (CLUR), especially in nonstationary conditions characteristic for LEO and HEO sensors. Furthermore, for very dim objects, the conventional tracking *after* detection techniques such as Extended Kalman Filter (EKF) and its modifications often perform poorly, as we will see below. In such cases, it is imperative to develop efficient real or near-real time tracking *before* detection methods.

This paper continues the work started in [7, 8] with the main focus on the development and evaluation of efficient CLUR algorithms for nonstationary clutter, including parallax effects for 3D scene content (terrain and clouds) observed from LEO orbits, as well as nonlinear filtering based *track-before-detect* algorithms for tracking multiple very dim point objects in highly stressing cluttered environments. The main contribution is applying CLUR and track-before-detect algorithms jointly in difficult conditions when conventional approaches completely fail. The algorithms are designed to handle a wide variety of real-world challenges, including significant celestial, earth limb and terrestrial clutter. The performance of the developed algorithms is demonstrated using synthetic data containing real-world

Address correspondence to A. G. Tartakovsky, Argo Science Corp., 71 Cypress Way, Rolling Hills Estates, CA 90274; E-mail: tartakov@argoscience.com or A. P. Brown, Toyon Research Corporation, 6800 Cortona Drive, Goleta, CA 93117; E-mail: abrown@toyon.com

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **SEP 2010**	2. REPORT TYPE	3. DATES COVERED **00-00-2010 to 00-00-2010**

4. TITLE AND SUBTITLE
Nonstationary EO/IR Clutter Suppression and Dim Object Tracking

5a. CONTRACT NUMBER
5b. GRANT NUMBER
5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER
5e. TASK NUMBER
5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory,Hanscom AFB,MA,01731

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR’S ACRONYM(S)

11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
2010 Advanced Maui Optical and Space Surveillance Technologies Conference, 14-17 Sep, Maui, HI.

14. ABSTRACT
We develop and evaluate the performance of advanced algorithms which provide significantly improved capabilities for automated detection and tracking of ballistic and flying dim objects in the presence of highly structured intense clutter. Applications include ballistic missile early warning, midcourse tracking, trajectory prediction, and resident space object detection and tracking. The set of algorithms include, in particular, adaptive spatiotemporal clutter estimation-suppression and nonlinear filtering-based multiple-object track-before-detect. These algorithms are suitable for integration into geostationary, highly elliptical, or low earth orbit scanning or staring sensor suites, and are based on data-driven processing that adapts to real-world clutter backgrounds, including celestial, earth limb, or terrestrial clutter. In many scenarios of interest, e.g., for highly elliptic and, especially, low earth orbits, the resulting clutter is highly nonstationary, providing a significant challenge for clutter suppression to or below sensor noise levels which is essential for dim object detection and tracking. We demonstrate the success of the developed algorithms using semi-synthetic and real data. In particular, our algorithms are shown to be capable of detecting and tracking point objects with signal-to-clutter levels down to 1/1000 and signal-to-noise levels down to 1/4.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**10**	

clutter scenes, with synthetic objects and focal plane array noise, allowing performance characterization at different signal-to-clutter and signal-to-noise ratios.

The paper is organized as follows. In Section 2, we outline the CLUR technique for highly nonstationary conditions that arise in scenarios where sensors are situated on moving platforms such as LEO and HEO satellites. Section 3 describes the nonlinear filtering based track-before-detect method. In Section 4, we illustrate efficiency of the developed CLUR and track-before-detect algorithms in stressing conditions. Section 5 concludes the paper.

2. ESTIMATION AND REJECTION OF NONSTATIONARY CLUTTER

We observe a sequence of $N_x \times N_y$ IR images (intensities) that have the form

$$I_n(i,j) = \sum_{k=1}^{K_n} S_n^k(i,j) + b_n(i,j) + \xi_n(i,j), \quad n \geqslant 1,\ i = 1,\dots,N_x,\ j = 1,\dots,N_y, \tag{1}$$

where $\xi_n(i,j)$ is sensor noise (in the n^{th} frame) which is usually white in space and time; $b_n(i,j)$ is clutter; $S_n^k(i,j)$ is a signal from the k^{th} object; K_n is an unknown number of objects in the n^{th} frame; (i,j) is the pixel in the plain image. Obviously, if there are no objects in the scene, the first term is absent.

We focus on the residual-type CLUR algorithms of the form

$$\tilde{I}_n(i,j) = I_n(i,j) - \hat{b}_n(i,j), \quad n \geqslant 1, \tag{2}$$

where $\tilde{I}_n(i,j)$ is the output of the CLUR filter and $\hat{b}_n(i,j)$ is an estimate of clutter $b_n(i,j)$ in the pixel (i,j) of the current n^{th} frame based on the previous frames $I_{n-m+1}(i,j),\dots,I_n(i,j)$ in a sliding window of the size m ($n \geqslant m$). In other words, background clutter is first estimated using spatiotemporal image processing and then subtracted from the observed image intensity. It follows from (1) and (2) that the images at the output of the CLUR algorithm have the form

$$\tilde{I}_n(i,j) = \sum_{k=1}^{K_n} \tilde{S}_n^k(i,j) + \tilde{\xi}_n(i,j), \quad n \geqslant 1,\ i = 1,\dots,N_x,\ j = 1,\dots,N_y, \tag{3}$$

where $\tilde{S}_n^k(i,j)$ are transformed signals due to filtering and $\tilde{\xi}_n(i,j)$ is the sum of the residual clutter (artifacts) and sensor noise. Clearly, if the estimate $\hat{b}_n(i,j)$ is accurate, the residual $\tilde{\xi}_n(i,j)$ will look like sensor noise $\xi_n(i,j)$, i.e., the output, clutter-suppressed frames will be whitened.

As we discussed in [8], in highly nonstationary conditions, the CLUR algorithm should take into account the following specific features:

- Nonlinearity of images.
- Nonstationarity related to sensor motion: not only translational and rotational stabilization is needed, but also image prediction (extrapolation), including compensation of perspective changes and parallax.
- Three-dimensional cloud cover.
- Complexity of images due to discontinuities.

The CLUR algorithm can be summarized as follows:

- The estimation of the projective transformation is performed based on a linear approximation of the image intensity.
- This results in a system of eight nonlinear equations solved by the Newton method.
- For estimating large projection distortions, a hierarchical (multi-layer) approach with Gaussian pyramids is used.
- The algorithm covers very general models with the images that are 3D-to-2D projections of a plane with an arbitrary orientation in 3D.

The basic steps of the CLUR algorithm have been described in our previous AMOS paper [8]. Here we only mention that for practical purposes the intensity of the image in the vicinity of each pixel is approximated by the linear model, i.e.,

$$I_n(x+dx, y+dy) \approx I_n(x,y) + \frac{\partial I_n}{\partial x} dx + \frac{\partial I_n}{\partial y} dy.$$

3. NONLINEAR FILTERING-BASED TRACK BEFORE DETECT

Traditional object trackers are based on the following sequence of operations. In each frame, thresholding of a detection metric (sensitive to changes in the distribution of the residual clutter and noise due to object presence) is performed, to detect potential objects. Detections are processed for track initialization, confirmation (or rejection) and termination by a track maintenance/management block that includes data association and filtering/trajectory estimation. Object state estimation based on the EKF (or variants) in combination with switching interacting multiple modeling for maneuvering objects, is common, as is data association based on nearest-neighbor, joint probabilistic, or multiple hypothesis algorithms. See, e.g., Bar-Shalom and Li [2] and Blackman and Popoli [3].

This approach works well for tracking bright and moderately intense objects, but it is not appropriate for tracking low-observable objects. There are at least three drawbacks of this approach. First, thresholding leads to loss of some information. Second, estimation methods based on linearization, like the EKF, become unstable for signal-to-noise ratio (SNR) values less than 3–6 dB. Optimal nonlinear filtering (ONLF) may partially solve the problem for low SNR values, down to 0–1 dB, but definitely not less. Third, data association ambiguities pose a major challenge in low-observable object tracking: in situations with severe clutter and/or noise, if the detection threshold is set low enough to detect very dim objects, the number of clutter/noise detections may result in either poor track estimation performance (if simple data association algorithms are used), or may prohibit real-time operation as the number of data association hypotheses explodes (if advanced data association algorithms, such as multiple hypothesis tracking, are used).

In contrast, for low-observable object tracking, we employ an approach known as *track-before-detect*, in which information from individual frames is integrated over time until the number and locations of objects can be accurately estimated. See, e.g., Reed, Gagliardi, and Stotts [5], Kligys, Rozovskii, and Tartakovsky [4], Ristic, Arulampalam, and Gordon [6]. In our approach to track-before-detect, which we develop in a Bayesian framework, clutter-suppressed frames are used to update a non-parametric statistical background scene model that is built up on-the-fly to account for intensity variations due to effects such as noise and clutter leakage. Each clutter-suppressed frame is evaluated in the statistical model to estimate the probability that the intensity of each optical sample has been modified by the presence of a moving object. The moving object probabilities are then integrated over time, based on probabilistic constraints on possible object motions, to develop the *a posteriori* probability distribution function over the space of possible object locations, based on the measurement history and any available *a priori* information (note that this *a priori* information is not a requirement).

The Bayesian framework for identification and filtering problems represents a theoretically sound approach which entails determining the probability density function (PDF) of an unknown parameter vector (in this case, a time-varying object state). The following formula, referred to as Bayes' Rule, provides a means by which the prior PDF of some unknown random time-varying parameter vector, $\mathbf{x}_k$, is updated using a likelihood function

$$p(\mathbf{x}_k \mid Z^{1:k}) = \frac{p(\mathbf{z}_k \mid \mathbf{x}_k) p(\mathbf{x}_k \mid Z^{1:k-1})}{p(\mathbf{z}_k \mid Z^{1:k-1})},$$

where $Z^{1:k} = (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_k)$ represents a vector of measurements up to time k; in this case, the history of clutter-suppressed images/frames $\mathbf{z}_n = \left\| \tilde{I}_n(i,j) \right\|$, $n = 1, \ldots, k$ through the current frame index, k (see (2) and (3)). The measurements are related to the unknown parameter through the following measurement equation:

$$\mathbf{z}_n = \mathbf{h}_n(\mathbf{x}_n, \mathbf{n}_n), \quad n \geqslant 1,$$

where $\mathbf{h}_n(\cdot,\cdot)$ is a possibly time-varying nonlinear function of the unknown parameter $\mathbf{x}_n$, and $\mathbf{n}_n$ is a sample of a measurement noise process. Usually, the PDF of the measurement process is assumed known; however, it is only required that the likelihood function $p(\mathbf{z}_n \mid \mathbf{x}_n)$ is known or can be estimated. In our case, *we estimate* $p(\mathbf{z}_n \mid \mathbf{x}_n)$ *using empirical statistics calculated from the clutter-suppressed frames*. The state is assumed to evolve according to the following general dynamic equation:

$$\mathbf{x}_{n+1} = \mathbf{f}_n(\mathbf{x}_n, \mathbf{v}_n), \quad n \geqslant 0,$$

where $\mathbf{f}_n(\cdot,\cdot)$ may be a time-varying nonlinear function of the state and $\mathbf{v}_n$ is typically modeled as a white noise process (used to probabilistically model object maneuvers) with a known PDF. The difficulty in analytically determining the PDF of $\mathbf{x}_n$ is that only for special cases (not the ones we are interested in here) does a closed-form (i.e., finite) description of $p(\mathbf{x}_k \mid Z^{1:k})$ exist. Monte Carlo/particle filter algorithms provide a means for approximating $p(\mathbf{x}_k \mid Z^{1:k})$, and are well-suited for handling nonlinearities in object dynamics and optical measurement (which involves a highly nonlinear projective function) models. In our real-time C++ software, we have implemented a sampling-importance-resampling (SIR)-based particle filter, for modeling and updating the posterior PDF [1]. In our software, a track can be initiated by an external cue, such as a track formed from another sensor processor, or initiated automatically by our multiple object hypothesis management algorithm. In the latter case, detections/tracks are declared and reported when likelihoods of hypothesized object trajectories become sufficiently large. Subsequently, the track state estimates are updated with information from each new image as it becomes available.

4. EXPERIMENTS

In order to evaluate operating characteristics of the developed CLUR and tracking algorithms, we performed extensive experiments with semi-real and real data sets. The focus in this paper is on new results for highly nonstationary clutter conditions, relevant to LEO and HEO-based sensing. We refer to Tartakovsky, Brown, and Brown [8] for previous results for the applications of geostationary-based sensing of scenes with terrestrial backgrounds, and LEO-based sensing of scenes with celestial backgrounds (containing resident space objects).

To guarantee flexibility, we have developed simulators which allow us to mimic real LEO and HEO scenarios using real clutter images (of terrain and clouds), and to insert objects with various intensities and trajectories (dynamics), as well as to simulate focal plane array noise at various signal-to-noise ratios. We have also developed an interface between CLUR algorithms and tracking algorithms, as well as a graphical user friendly interface (GUI) to input-output data and visualize the results of processing. Sample experiments are describes below.

4.1. CLUR and Conventional Tracking. In this subsection, we present the results of the CLUR algorithm implementation jointly with a conventional tracking after detection technique that first performs thresholding to detect potential objects in each frame, then initializes tracks over several frames, and finally maintains confirmed tracks over time. The "track-after-detect" technique is shown to not succeed for low SNR.

We performed a number of simulation experiments with semi-synthetic data. The test image with real intense clutter (clouds, earth surface, sea, and combinations thereof) were imbedded in the proprietary simulator that imitates real HEO and LEO orbits. Artificial point objects were inserted in the images prior to clutter suppression.

The first experiment regards the following HEO scenario:

- HEO Orbit Parameters: Perigee $= 500$ km; Apogee $= 35000$ km; Longitude of Ascending Node $= 0.01$ rad; Argument of Perigee $= 0.01$; Inclination $= 0.5$ rad;

- Camera Parameters (camera is always pointed to the image center): FOV Vertical = FOV Horizontal $= 0.012$ rad; Additional Jitter Amplitude $= 0.5$ pixel;

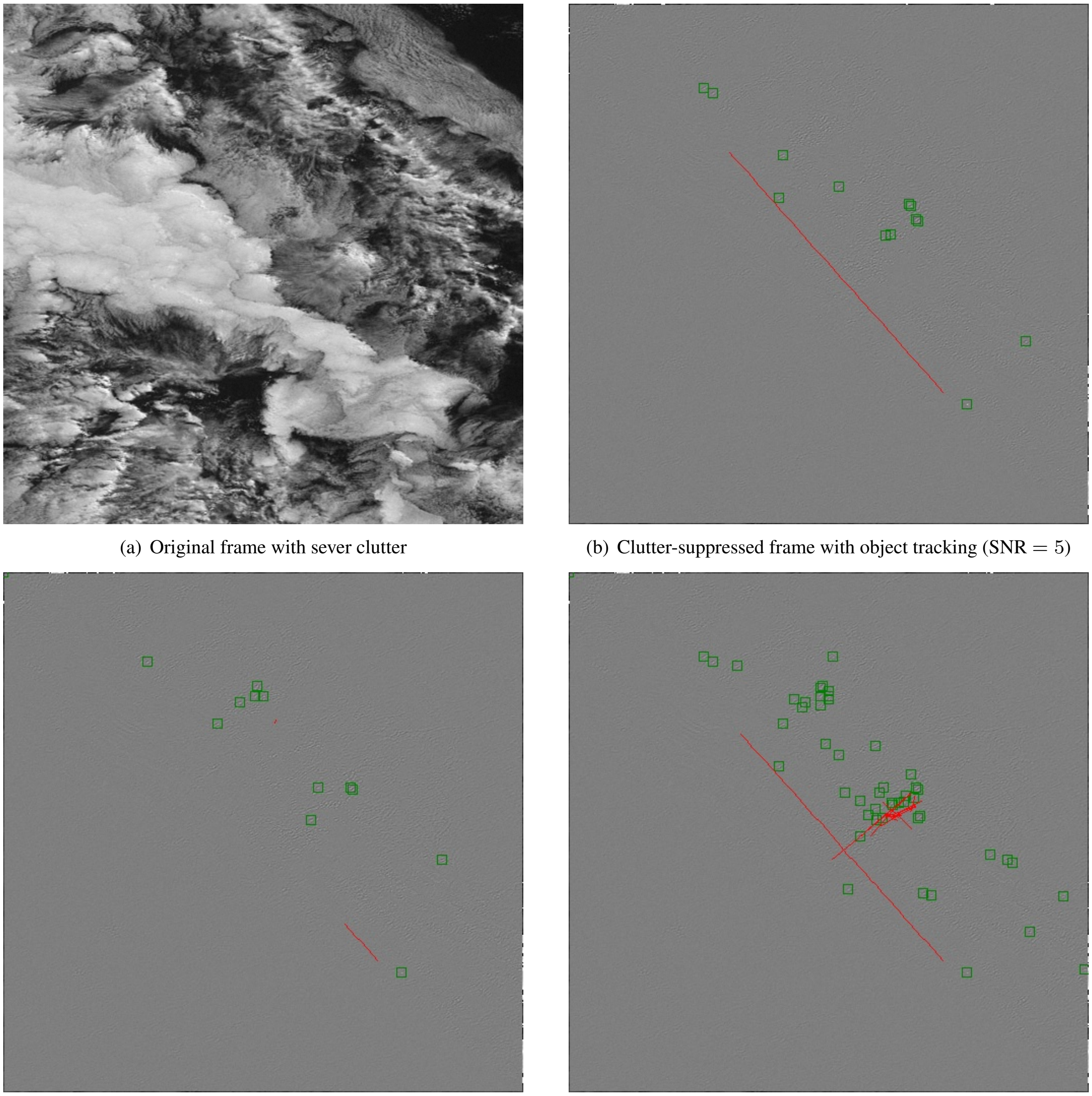
- Background Parameters: Image Location – Latitude $= -1$ rad, Longitude $= 1.5$ rad; Image Scale $= 1$ km/pixel; Image Size $= 512 \times 512$;

- Object Parameters (XY reference at center and axis looking down-right): Start Position $X_0 = Y_0 = -1600$ km; Speed $V_x = V_y = 70$ m/sec; Linear motion;

- CLUR Algorithm Parameters: CLUR Filter Memory $m = 10$ frames; Hierarchy (pyramid) level $= 6$; Maximum number of iterations $= 20$; Iterations threshold $= 3 \cdot 10^{-8}$.

Figure 1 illustrates the results of clutter suppression and object tracking in clutter-suppressed images for the nonstationary HEO data with the above parameters. In all cases the signal-to-clutter+noise-ratio was below 1/200, so that finding this point object in the cluttered image is similar to finding a needle in the stack. Despite this fact, when the object is relatively bright compared to sensor noise (signal-to-noise-ratio about 5), it is reliably tracked, as seen from Figure 1(b). This is because the spatiotemporal CLUR algorithm suppresses clutter to or even below the sensor noise level.

Our experiments show that with this sequence of images objects with lower signal-to-noise-ratios cannot be reliably tracked with conventional tracking methods. Figures 1(c) and 1(d) confirm this statement. When we used a relatively high detection threshold (corresponding to the probability of false alarm $\mathsf{PFA} = 5 \cdot 10^{-5}$), in which case false tracks are either not produced or produced with very small probability, the dim object was not tracked most of the time (only in the very end), as Figure 1(c) shows. It follows from Figure 1(d) that lowering the detection threshold (to achieve



(a) Original frame with sever clutter

(b) Clutter-suppressed frame with object tracking (SNR $= 5$)

(c) Clutter-suppressed frame with tracking (SNR=1, high threshold)

(d) Clutter-suppressed frame with tracking (SNR=1, low threshold)

Figure 1. Clutter suppression and object tracking with a detect-then-track algorithm for the HEO scenario. Red lines depict object tracks, green boxes – instantaneous detections.

$\mathsf{PFA} = 2 \cdot 10^{-4}$) allowed us to track this object, but at the expense of multiple false tracks that form a branching processes. If the brightness of the object is further reduced, the track-after-detect algorithm does not track the object.

The second experiment regards the following LEO scenario:

- LEO Orbit Parameters: Perigee $= 200$ km; Apogee $= 400$ km; Longitude of Ascending Node $= 6.26$ rad; Argument of Perigee $= 0$; Inclination $= 0.02$ rad;

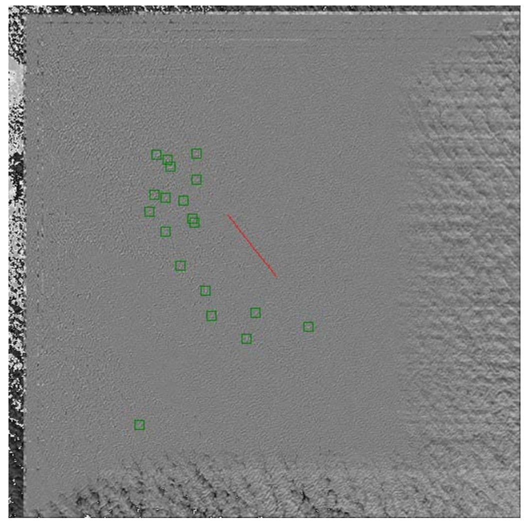
- Camera Parameters (camera is always pointed to the image center): FOV Vertical $=$ FOV Horizontal $= 0.02$ rad; Additional Jitter Amplitude $= 0.5$ pixel;

- Background Parameters: Image Location – Latitude $=$ Longitude $= 0$ rad; Image Scale $= 0.01$ km/pixel; Frame size $= 512 \times 512$;

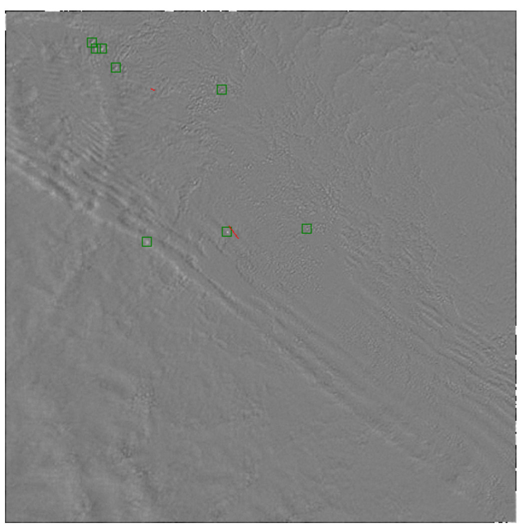
- Object Parameters (XY reference at center and axis looking down-right): Start Position $X_0 = Y_0 = -0.7$ km; Speed $V_x = V_y = 30$ m/sec; Linear motion;
- CLUR Algorithm Parameters: CLUR Filter Memory $m = 10$ frames; Hierarchy (pyramid) level $= 6$; Maximum number of iterations $= 20$; Iterations threshold $= 3 \cdot 10^{-8}$.

Figures 2(a), 2(b), and 2(c) correspond to this scenario. The conclusion is analogous to the previous HEO case: when SNR ≈ 6 the object can be reliably tracked with traditional tracking (after detection) algorithms. However, dim objects cannot be tracked. This is not surprising since it is known that the standard tracking methods such as EKF and its modifications become unstable for SNR levels below 3dB. Therefore, even an ideal CLUR algorithm that suppresses clutter to almost zero cannot help in such cases.

(a) Original frame with sever clutter



(b) Clutter-suppressed frame with object tracking (SNR = 6)

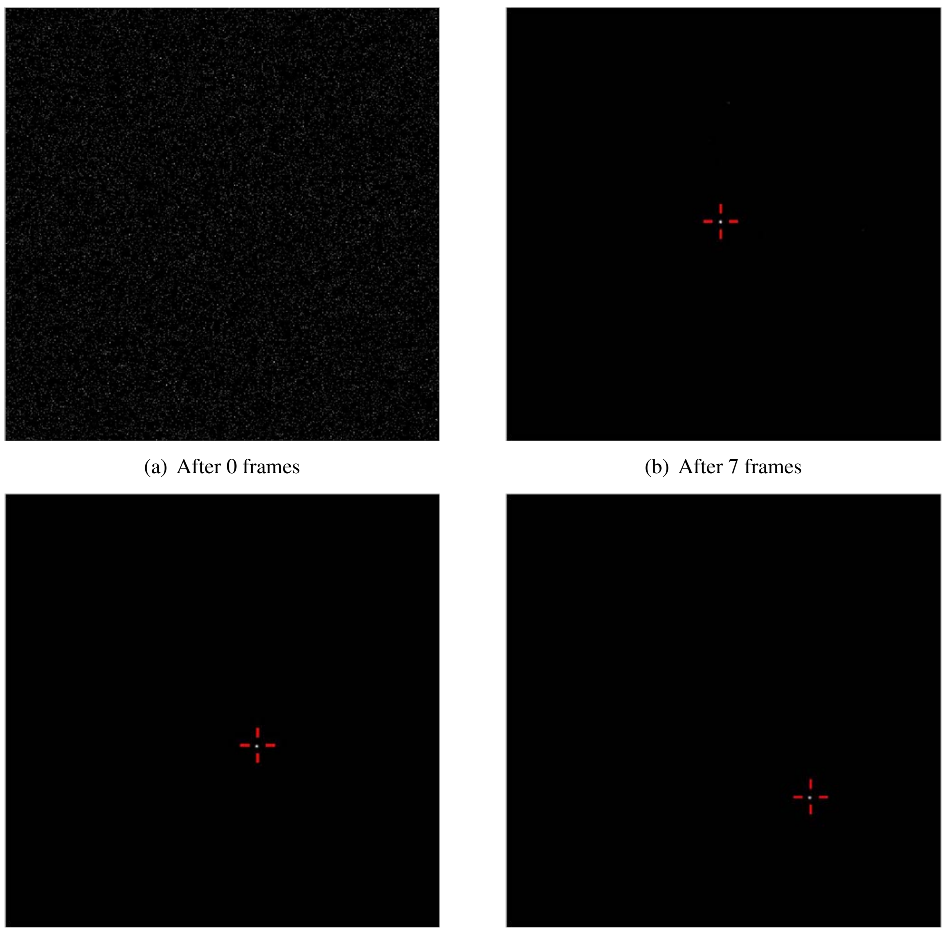


(c) Clutter-suppressed frame with object tracking (SNR = 1)

Figure 2. Clutter suppression and object tracking with a detect-then-track algorithm for the LEO scenario. Red line depicts object track, green boxes – instantaneous detections.

Thus, we can conclude that for detecting and tracking dim objects a novel approach is needed. It turns out that the track-before-detect method described in Section 3 is a panacea. The corresponding results are presented in the next subsection.

4.2. CLUR and Track-Before-Detect for LEO and HEO. To demonstrate the benefit of combined CLUR and track-before-detect in the low observable object HEO case of Figure 1, where classical detect-then-track processing failed, we processed the sequence of clutter-suppressed frames using our particle filter-based nonlinear track-before-detect algorithm, and achieved the results shown in Figure 3. As shown in the four sub-figures, the particle distribution (initially uniform in the image plane, in (a)) quickly converges to the object location, and the object hypothesis management algorithm jointly confirms detection and the object track after processing just 7 frames, in (b). The red cross hairs provide a visual cue to the location of the object in track, and the track is maintained through the remaining frames of the sequence, as shown in (c)–(d). In this case, tracking was performed in the image plane, and a 2D white noise acceleration object dynamics model was used, with the following parameters: mean object speed $= 10$ pixels/sec, standard deviation of object speed $= 2.5$ pixels/sec, and standard deviation of object acceleration $= 1$ pixel/sec. In the particle filter, the number of particles was set to 100,000 (currently runs at approximately 0.75 fps using a single micro-processor core). We believe that a smaller number of particles could have been used, and faster run times achieved. For example, next we show the ability to track a simulated midcourse ballistic object observed from LEO at signal-to-clutter ratios down to 1/1000 using 20,000 particles.



(a) After 0 frames

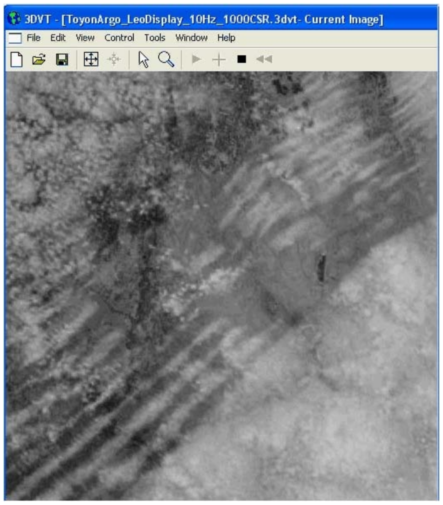
(b) After 7 frames

(c) After 14 frames

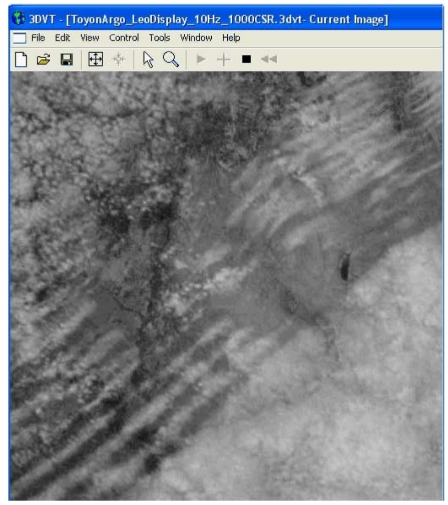
(d) After 21 frames

Figure 3. Example results of integrated CLUR and track-before-detect for the low observable object HEO case of Figure 1. The images shown represent the output of the particle filter, with lighter intensities (more particles within an optical sample) corresponding to more likely object locations. The red cross hairs indicate the estimated location of the object after track confirmation.

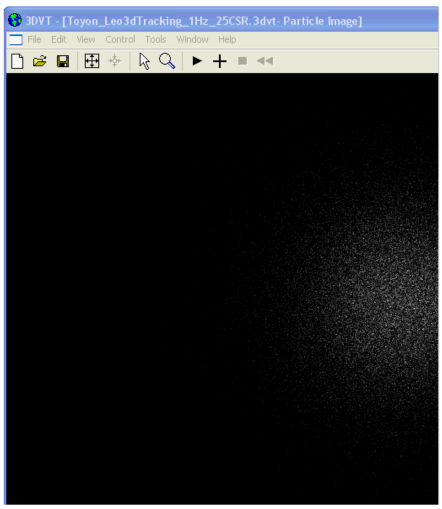
In Figure 4, we present 3D track-before-detect results for a LEO simulation including realistic parallax effects. The simulation was developed in a Toyon GIS application using the OpenGL programming environment to create the



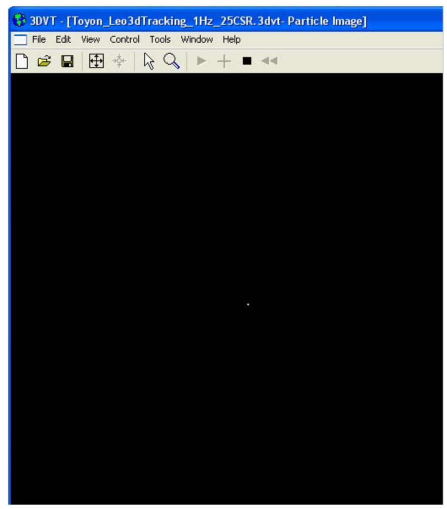
(a) First frame ($t = 0$ sec.)



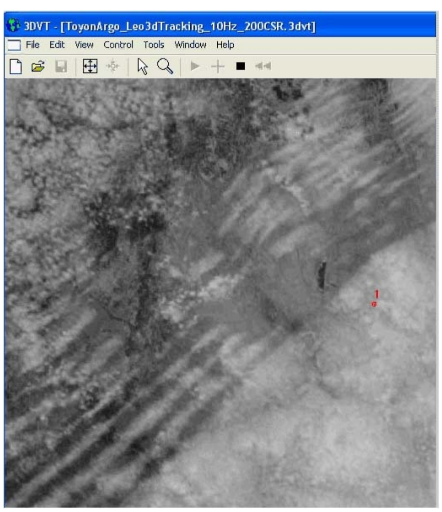
(b) Last frame ($t = 30$ sec.)



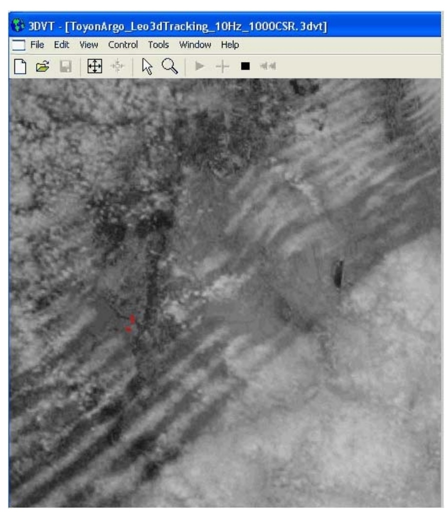
(c) Initial particle distribution from track handoff



(d) Particle distribution after convergence and detection/track confirmation



(e) Visual cue indicating initial detection/track confirmation for SCR $= 1/200$



(f) Visual cue indicating initial detection/track confirmation for SCR $= 1/1000$

Figure 4. LEO scenario simulation with realistic parallax effects.

3D models and motion of all objects in the system, including the objects, satellite, and rotating earth, including both terrain and clouds (modeled as planes located at specified altitudes above the terrain), enabling simulation of complex parallax effects (including occlusion/un-occlusion of the terrain by clouds) due to the sensor motion. In the scenario shown in the figure, the object was a single ballistic missile covering a ground distance of 750 km. The missile is of sub-pixel size, but due to its high velocity, its energy was blurred over 2 pixels during integration of each image. Each pixel is 170 meters wide (ground sample distance), and the images are of size 512×512 pixels, collected from an altitude of 1,000 km, with a 5-deg.-wide field of view (search/acquisition mode) at 10 Hz. Image sequences were generated with different object intensities to allow characterization of algorithm performance vs. signal-to-clutter-ratios (SCRs), and different FPA additive noise realizations were generated, at various SNRs. The GIS application allows the simulation to be viewed from an inertial viewpoint looking at all objects in the scene, as well as from the intended viewpoint of the satellite, with the sensor perspective focused in any desired direction. Additionally, multiple satellites and missiles may exist simultaneously during a simulation, allowing the scene to be viewed from different perspectives at the same moment in time. In (a)–(b), two example images collected 30 sec. apart are shown: note the image plane motion of the clouds, relative to the terrain, due to the change in sensor location/perspective. The images in this sequence were rendered from two measured clutter images, one terrestrial and one of cloud cover, with the altitude of the clouds above the surface of the terrain set to 9 km.

For the scenario shown in Figure 4, tracking was performed in 3D ECEF coordinates, using 20,000 particles. The initial particle cloud was generated from a simulated track handoff (e.g., from plume detection during launch or a subsequent radar track) distributed with a ± 3-sigma diameter of approximately 60 km (area of approximately 50,000 pixels in the image plane projection). For a case where SCR $= 1/25$ and SNR $= 1/4$ (-6 dB), the particle distribution is shown in (c)–(d). After multiple frames, the particle distribution converges such that clusters of particles are located in the most likely regions of the state space, with a small fraction of the particles distributed more diffusely. As a result, we have found that our particle filter-based track-before-detect algorithm is more than two orders of magnitude more efficient (in terms of computation and storage requirements) compared with the classical 3D velocity-matched filter [5], for the same level (near-optimal effective SNR increase) of performance. This is true even though the particle filter was configured to account for possibly large object maneuver accelerations. The explanation for the efficiency gain in the recursive particle filter algorithm is that after processing each frame of data, during the particle re-sampling step, the particles are automatically re-allocated to perform more finely-spaced searches for object locations and velocities in regions of the state space that are found to be most likely (based on processing multiple past frames of data), whereas in the 3D velocity-matched filter, hypotheses are allocated uniformly while processing the entire batch of frames. In (e)–(f), we show joint detection/track confirmation results for cases where the SNR was 4/1, and SCRs 1/200 (e) and 1/1000 (f) were considered. Visual track cues (ellipse indicates 3-sigma uncertainty and centroid indicates track state estimate) are overlaid on the images. Note that track confirmation occurs later for lower SCRs at a given frame rate, as expected.

5. CONCLUSION

While intuitively obvious, the results of this paper show explicitly that reliable tracking dim, small objects in nonstationary clutter is impossible without sophisticated spatiotemporal processing providing accurate clutter estimation and rejection to or below sensor noise levels, which we demonstrate for both HEO and LEO scenarios. Furthermore, in many difficult low-observable object scenarios, classical tracking after detection methods are not efficient enough to detect and track objects of interest even after clutter removal. We showed that an adequate solution can be built using a track-before-detect technology based on optimal nonlinear filtering. When combined with powerful spatiotemporal clutter rejection methods, the developed track-before-detect algorithms allow one to track very dim objects with exceptionally low signal-to-clutter ratios (down to 1/1000, or lower) and signal-to-noise ratios (down to 1/4, or lower). In addition to being very efficient in terms of operating characteristics (false alarm rate, accuracy of tracking, etc.), the proposed clutter rejection and track-before-detect algorithms can be implemented in near real time on a standard machine, such as desktop PCs. We have developed such an implementation in object-oriented C++ software. We believe that the developed technology will be useful for a variety of space-based EO and IR surveillance and early warning applications.

While further significant improvement of the developed track-before-detect algorithms for lower SCR and SNR conditions may not be possible, especially taking into account real-time implementation requirements, we believe that fruitful research could be conducted in the area of tracking more than a small number of closely-spaced objects. We leave this important practical problem for future research.

ACKNOWLEDGEMENTS

The work of Alexander Tartakovsky was supported in part by the Missile Defense Agency contract W91260-08-C-0004 at Argo Science Corp., as well as by the U.S. Army Research Office MURI grant W911NF-06-1-0094, and by the U.S. National Science Foundation grant CCF-0830419 at the University of Southern California. The work of Andrew Brown was supported in part by the Missile Defense Agency contract W91260-08-C-0004 at Toyon Research Corporation. The work of James Brown was supported by the Air Force.

We thank Alexei Khavaev for improvement and implementation of the spatiotemporal CLUR algorithm; Mark Meloon, Craig Agate, Robert Wilkerson, and Pete Stieber for their developments in track-before-detect; and Katie Resner and Pete Stieber for their developments in imagery simulation.

REFERENCES

[1] Arulampalam, M. S., Maskell, S., Gordon, N., and Clapp, T., A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking, *IEEE Transactions on Signal Processing*, Vol. 50, No. 2, 2002.

[2] Bar-Shalom, Y. and Li, X. R., *Estimation and Tracking: Principles, Techniques and Software*, Artech House, Boston-London, 1993.

[3] Blackman, S. and Popoli, R., *Design and Analysis of Modern Tracking Systems*, Artech House, Boston-London, 1999.

[4] Kligys, S., Rozovsky, B. L., and Tartakovsky, A. G., Detection Algorithms and Track Before Detect Architecture Based on Nonlinear Filtering for Infrared Search and Track Systems, *Technical Report # CAMS–98.9.1, Center for Applied Mathematical Sciences, University of Southern California*, 1998. (Available at http://www.usc.edu/dept/LAS/CAMS/usr/facmemb/ tartakov/preprints.html).

[5] Reed, I., Gagliardi, R., and Stotts, L., Optical Moving Target Detection with 3-D Matched Filtering, *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 24, No. 4, 1988.

[6] Ristic, B., Arulampalam, S., and Gordon, N., *Beyond the Kalman Filter-Particle Filters for Tracking Applications*, Artech House, 2004.

[7] Tartakovsky, A. G. and Brown, J., Adaptive Spatial Temporal Filtering Methods for Clutter Removal and Target Tracking, *IEEE Aerospace and Electronic Systems*, Vol. 44, No. 4, pp. 1522-1537, 2008.

[8] Tartakovsky, A. G., Brown, A., and Brown, J., Enhanced Algorithms for EO/IR Electronic Stabilization, Clutter Suppression, and Track-Before-Detect for Multiple Low Observable Targets, *Proceedings of the 10th Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS)*, CD-Rom, Maui, HW, September 1-4, 2009.